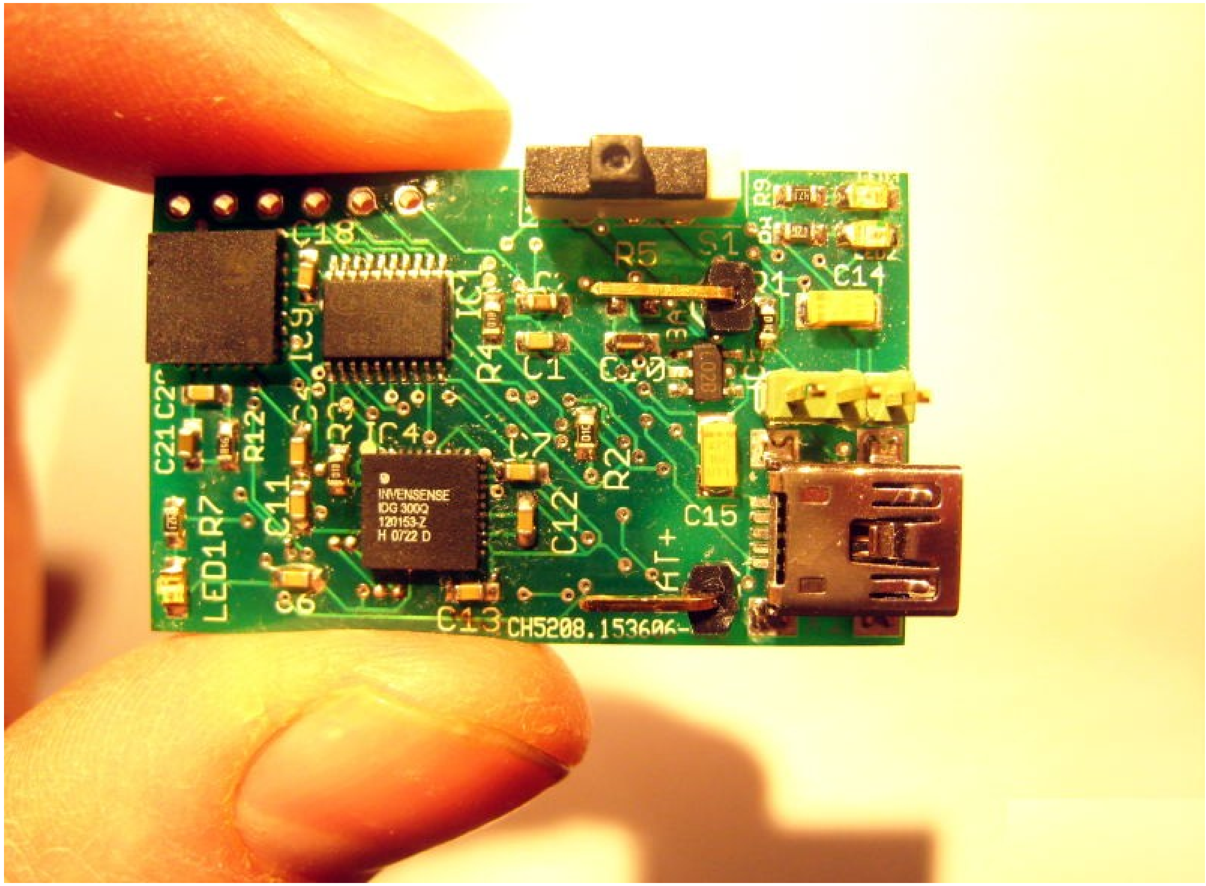
Bowsense - A minimalistic Approach to Wireless Motion Sensing

Hans Wilmers
Norwegian Centre for Technology in Music and the Arts
(NOTAM)
hans.wilmers@notam02.no



Soundprocessing, how we do it in Norway

The Sensorbow project at NMH

- Research project at the Norwegian Academy of Music together with NOTAM, with the aim to enhance string instruments by motion sensing of the bow.
- As part of the project, a sensorbow was developed at NOTAM that contains accelerometers, gyroscopes, a finger pressure sensor and 2 buttons.

➔ Bowsense, as a spinoff, is hopefully useful in other projects as well.

Other Bow Projects

- Hyperbow (MIT / Diana Young)
- Augmented Violin Project (IRCAM / Bevilacqua)
- K-Bow (Keith McMillen)

... some drawbacks in all of them ...

➔ We needed to develop ourselves.

What we needed

- light & small
- wireless
- measure acceleration and angular rate
- expandable with external sensors
- simple design

What we made

- light & small: 40x20mm, 12g including battery
- wireless
- measure acceleration and angular rate
- expandable with external sensors
- simple design

What we made

- light & small: 40x20mm, 12g including battery
- wireless: Bluetooth
- measure acceleration and angular rate
- expandable with external sensors
- simple design

What we made

- light & small: 40x20mm, 12g including battery
- wireless: Bluetooth
- 3D accelerometer + 3D gyroscope
- expandable with external sensors
- simple design

What we made

- light & small: 40x20mm, 12g including battery
- wireless: Bluetooth
- 3D accelerometer + 3D gyroscope
- expandable with 4 external sensors
- simple design

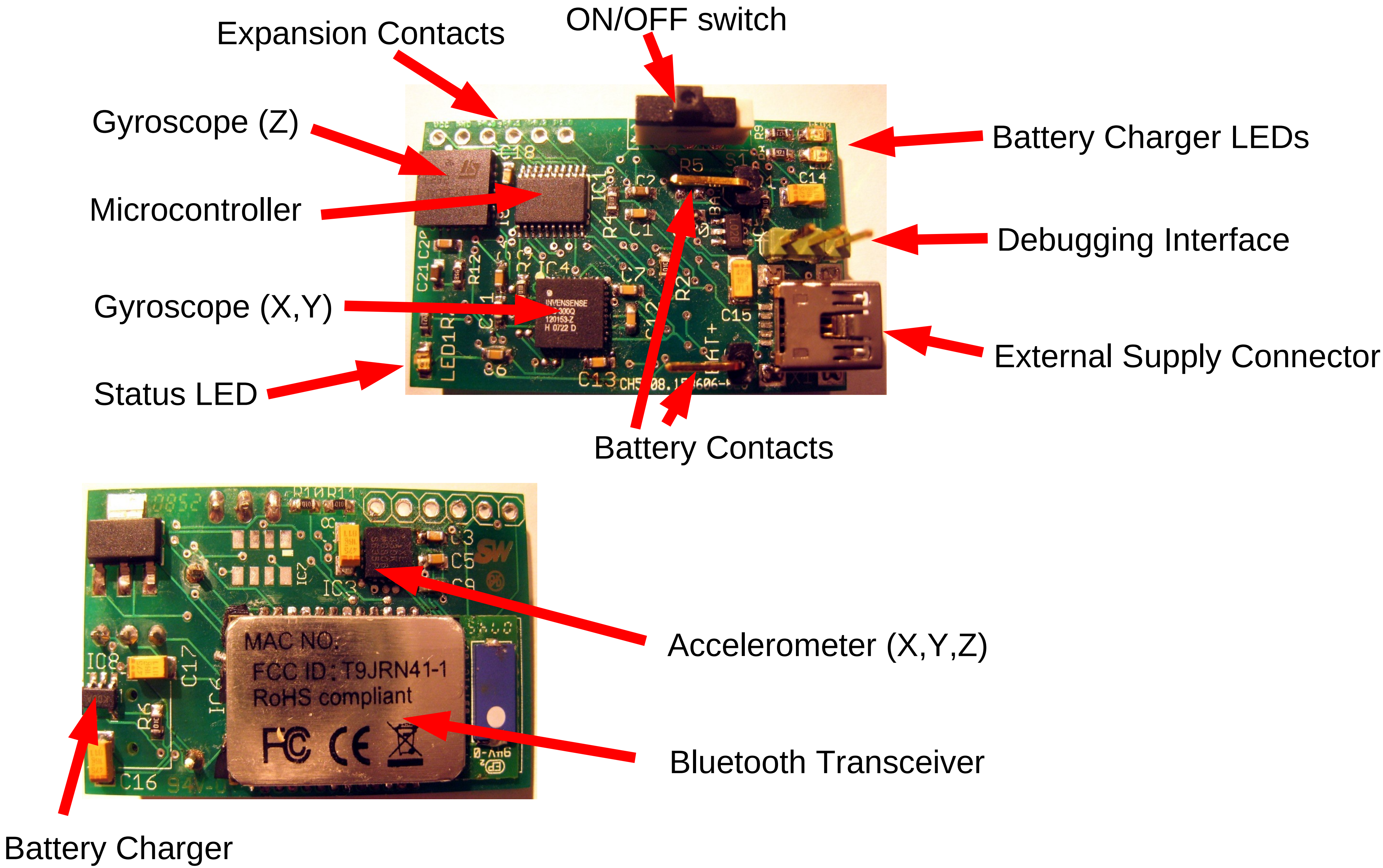
What we made

- light & small: 40x20mm, 12g including battery
- wireless: Bluetooth
- 3D accelerometer + 3D gyroscope
- expandable with 4 external sensors
- simple design: Microcontroller contains 12bit A/D converter, no OpAmps needed, hand solderable

What we made

- light & small: 40x20mm, 12g including battery
- wireless: Bluetooth
- 3D accelerometer + 3D gyroscope
- expandable with 4 external sensors
- simple design: Microcontroller contains 12bit A/D converter, no OpAmps needed, hand solderable
- open hard- and software

The Hardware



Hardware Characteristics

Functionality	Component	Data
Microcontroller	C8051F530 (Silabs)	256Byte RAM, 8kByte Flash, 12bit A/D, 25MIPS
Accelerometer	ADXL330 (Analog Devices)	range +- 3g
Gyroscopes	X/Y: IDG300 (Invensense) Z: LISY300AL (ST)	range +- 500°/s range +- 300°/s
Bluetooth	RN41 (Roving Networks)	range <= 100m
Battery	LPP402025 (Varta)	Li-Polymer, 150mAh capacity

Runtime from fully charged battery: 2..2.5h

Firmware

... consists of:

- Measurement of sensor voltages
- Scaling to physical unities
- Gravity estimator
- Data transfer into bluetooth module

Gravity Estimator

- implemented using 32bit integer arithmetics
- runs on bowsense (8bit microcontroller) in realtime
- simpler than a Kalman Filter
- allows auto calibration of angular rate

Gravity Estimator

- Rotation matrix for rotation around X axis:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{pmatrix}$$

... converges for small rotations to:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{pmatrix}$$

Gravity Estimator

- Small rotation around 3 axes:

$$R = \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}$$

Gravity Estimator

- Now use this in an iterative calculation:

$$\vec{g}' = \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} \vec{g} + \epsilon \vec{a}$$

(the term $\epsilon\vec{a}$ has the effect of a lowpass, and attracts $\vec{g}$ to $\vec{a}$)

Gravity Estimator

- Instead of a lowpass, add to rotation:

$$\vec{g}' = \begin{pmatrix} 1 & -(\gamma + \epsilon_3) & (\beta + \epsilon_2) \\ (\gamma + \epsilon_3) & 1 & -(\alpha + \epsilon_1) \\ -(\beta + \epsilon_2) & (\alpha + \epsilon_1) & 1 \end{pmatrix} \vec{g}$$

with $(\epsilon_1, \epsilon_2, \epsilon_3) =$ small rotation towards $\vec{a}$

Gravity Estimator

- Now, do this:

$$\vec{g}' = \begin{pmatrix} 1 & -(\gamma + \epsilon_3) & (\beta + \epsilon_2) \\ (\gamma + \epsilon_3) & 1 & -(\alpha + \epsilon_1) \\ -(\beta + \epsilon_2) & (\alpha + \epsilon_1) & 1 \end{pmatrix} \vec{g}$$

... and scale $\vec{g}$ to unity length.

Calibration

- Angular Rate:
  - offset is removed by lowpass
  - scale is adjusted by gravity estimator

- Acceleration:
  - offset is removed by attracting the acceleration vector to a sphere with radius=1g
  - scale is hardcoded

Data Transfer

- simple binary protocol, fixed packet length:

Header	Data	Footer
0x23 0x40	word0, word1,...	0x24

- sent via Bluetooth Serial Port Profile (SPP)
- received via rfcomm device
- converted to OSC by standalone app (readbow)

Connecting bowsense to the App

via OSC ...

bowsense

(bluetooth)

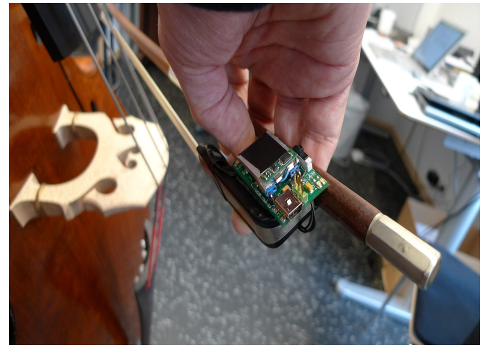
rfcomm

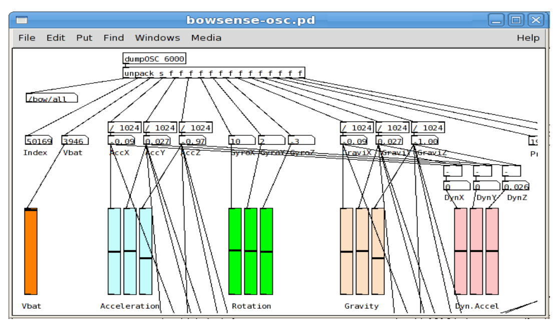
(serial)

readbow

(OSC)

Application





... or directly

bowsense

(bluetooth)

rfcomm

(serial)

Application

Three words about Bluetooth

- the good:
  - standardised and readily available
  - supported by all major operating systems
  - theoretical range 100m
- the bad:
  - range depends on data rate
  - works differently on Windows, OSX and Linux
- the ugly:
  - unstable at long range
  - needs long time to start a connection

Is open hardware free?

- The step between design and the actual product is steeper for hardware than for software. So how free can hardware get?
- Hardware design documents can be licensed like software, but hardware design itself can only be protected by patents. So how can we invite to collaboration?

➔ How to license open hardware?

What to do next?

- evaluate alternatives to bluetooth
  (e.g. Zigbee, WLAN)
- improve receiver software
- evaluate other sensors
- use with other musical instruments
- investigate other uses

Any Questions?